Exploring Supervised and Unsupervised Rewards in Machine Translation

Julia Ive[1], **Zixu Wang**[1], **Marina Fomicheva**[2], **Lucia Specia**[1,2,3]
Imperial College London[1], University of Sheffield[2], ADAPT - Dublin City University[3]
j.ive@ic.ac.uk, zixu.wang@imperial.ac.uk
m.fomicheva@sheffield.ac.uk, l.specia@ic.ac.uk

Abstract

Reinforcement Learning (RL) is a powerful framework to address the discrepancy between loss functions used during training and the final evaluation metrics to be used at test time. When applied to neural Machine Translation (MT), it minimises the mismatch between the cross-entropy loss and non-differentiable evaluation metrics like BLEU. However, the suitability of these metrics as reward function at training time is questionable: they tend to be sparse and biased towards the specific words used in the reference texts. We propose to address this problem by making models less reliant on such metrics in two ways: (a) with an entropy-regularised RL method that does not only maximise a reward function but also explore the action space to avoid peaky distributions; (b) with a novel RL method that explores a dynamic unsupervised reward function to balance between exploration and exploitation. We base our proposals on the Soft Actor-Critic (SAC) framework, adapting the off-policy maximum entropy model for language generation applications such as MT. We demonstrate that SAC with BLEU reward tends to overfit less to the training data and performs better on out-of-domain data. We also show that our dynamic unsupervised reward can lead to better translation of ambiguous words.

1 Introduction

Autoregressive sequence-to-sequence (seq2seq) neural architectures have become the *de facto* approach in Machine Translation (MT). Such models include Recurrent Neural Networks (RNN) (Sutskever et al., 2014; Bahdanau et al., 2014) and Transformer networks (Vaswani et al., 2017), among others. However, these models have as a serious limitation the discrepancy between their training and inference time regimes. They are traditionally trained using the Maximum Likelihood Estimation (MLE), which aims to maximise log-likelihood of a categorical ground truth distribution (samples in the training corpus) using loss functions such as cross-entropy, which are very different from the evaluation metric used at inference time, which generally compares string similarity between the system output and reference outputs. Moreover, during training, the generator receives the ground truth as input and is trained to minimise the loss of a single token at a time without taking the sequential nature of language into account. At inference time, however, the generator will take the previous sampled output as the input at next time step, rather than the ground truth word. MLE training thus causes: (a) the problem of "exposure bias" as a result of recursive conditioning on its own errors at test time, since the model has never been exclusively "exposed" to its own predictions during training; (b) a mismatch between the training objective and the test objective, where the latter relies on evaluation using discrete and non-differentiable measures such as BLEU (Papineni et al., 2002).

The current solution for both problems is mainly based on Reinforcement Learning (RL), where a seq2seq model (Sutskever et al., 2014; Bahdanau et al., 2014) is used as the policy which generates actions (tokens) and at each step receives rewards based on a discrete metric taking into account importance of immediate and future rewards. However, RL methods for seq2seq MT models also have their challenges: high-dimensional discrete action space, efficient sampling and exploration, choice of baseline reward, among others (Choshen et al., 2020). The typical metrics used as rewards (e.g., BLEU) are often biased and sparse. They are measured against one or a few human references and do not take into account alternative translation options that are not present in the references.

One way to address this problem is to use

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1908–1920
April 19 - 23, 2021.

entropy-regularised RL frameworks. They incorporate the entropy measure of the policy into the reward to encourage exploration. The expectation is that this leads to learning a policy that acts as stochastically as possible while able to succeed at the task. Specifically, we focus on the Soft Actor-Critic (SAC) (Haarnoja et al., 2018a,b) RL framework, which to the best of our knowledge has not yet been explored for MT, as well as other natural language processing (NLP) tasks. The main advantage of this architecture, as compared to other entropy regularised architectures (Haarnoja et al., 2017; Ziebart et al., 2008), is that it is formulated in the off-policy setting that enables reusing previously collected samples for more stability and better exploration. We demonstrate that SAC prevents the model from overfitting, and as a consequence leads to better performance on out-of-domain data.

Another way to address the problem of sparse or biased reward is to design an unsupervised reward. Recently, in Robotics, SAC has been successfully used in unsupervised reward architectures, such as the "Diversity is All You Need" (DIAYN) framework (Eysenbach et al., 2018). DIAYN allows the learning of latent-conditioned sub-policies ("skills") in unsupervised manner, which allows to better explore and model target distributions. Inspired by this work, we propose a formulation of an unsupervised reward for MT. We thoroughly investigate effects of this reward and conclude that it is useful in lexical choice, particularly the rare sense translation for ambiguous words.

Our **main contributions** are thus twofold: (a) the re-framing of the SAC framework such that it can be applied to MT and other natural language generation tasks (Section 3). We demonstrate that SAC results in improved generalisation compared to the MLE training, leading to better translation of out-of-domain data; (b) the proposal of a dynamic unsupervised reward within the SAC framework (Section 3.4). We demonstrate its efficacy in translating ambiguous words, particularly the rare senses of such words. Our datasets and settings are described in Section 4, and our experiments in Section 5.

2 Related Work

Reinforcement Learning for MT RL has been successfully applied to MT to bridge the gap between training and testing by optimising the sequence-level objective directly (Yu et al., 2017; Ranzato et al., 2015; Bahdanau et al., 2016). However, thus far mainly the REINFORCE (Williams, 1992) algorithm and its variants have been used (Ranzato et al., 2015; Kreutzer et al., 2018). These are simpler algorithms that handle the large natural language action space, but they employ a sequence-level reward which tends to be sparse.

To reduce model variance, Actor-Critic (AC) models consider the reward at each decoding step and use the Critic model to guide future actions (Konda and Tsitsiklis, 2000). This approach has also been explored for MT (Bahdanau et al., 2016; He et al., 2017). However, more advanced AC models with Q-Learning are rarely applied to language generation problems. This is due to the difficulty of approximating the Q-function for the large action space. The large action space is one of the bottleneck for RL for text generation in general. Pre-training of the agent parameters to be close to the true distribution is thus necessary to make RL work (Choshen et al., 2020). Further RL training of the agent makes the overfitting problem even more pronounced resulting in peaky distributions. Such problems are traditionally addressed by entropy regularised RL.

Entropy Regularised RL The main goal of this type of RL is to learn an efficient policy while keeping the entropy of the agent actions as high as possible. The paradigm promotes exploration of actions, suppresses peaky distributions and improves robustness. In this work, we explore the effectiveness of the maximum entropy SAC framework (Haarnoja et al., 2018a).

The work closest to ours is of Dai et al. (2018) where the Entropy-Regularised AC (ERAC) model leads to better MT performance. The major difference between ERAC and SAC is that the former is an on-policy model and the latter is an off-policy model. On-policy approaches use consecutive samples collected in real-time that are correlated to each other. In the off-policy setting, our SAC algorithm uses samples from the memory that are taken uniformly with reduced correlation. This key characteristic of SAC ensures better model generalisation and stability (Mnih et al., 2015). There are also differences in the architectures of SAC and ERAC, i.a., using 4 Q-value networks instead of two. These differences will be covered in detail in Section 3.

Unsupervised reward RL Significant work has been done in Robotics to improve the learning capability of robots. These approaches do not rely on a single objective but rather promote intrinsic motivation and exploration. Such an approach to learn diverse skills (latent-conditioned sub-policies, in practice, skills like walking or jumping) in unsupervised manner was recently proposed by Eysenbach et al. (2018). The approach relies on the SAC model and inspired our approach to designing our unsupervised reward for MT. We are not aware of other attempts to design dynamic unsupervised RL rewards (learnt together with the network) in seq2seq in general, or MT in particular. Recent work on unsupervised rewards in NLP (Gao et al., 2020) explores mainly static rewards computed against synthetic references.

3 Methodology

In this section we start by describing the underlying MT architecture and its variant using RL, to then introduce our SAC formulation and the reward functions used.

3.1 Neural Machine Translation (NMT)

A typical Neural Machine Translation (NMT) system is a seq2seq architecture (Sutskever et al., 2014; Bahdanau et al., 2014), where each source sentence $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is encoded by the encoder into a series of hidden states. At each decoding step t, a target word y_t is generated according to $p(y_t|y_{<t}, x)$ conditioned on the input sequence x and decoded sequence $\mathbf{y}_{<t} = (y_1, \cdots, y_{t-1})$ up to the t-th time step. Given the corpus of pairs of source and target sentences $\{x_i, y_i\}_{i=1}^{N}$, the training objective function - maximum likelihood estimation (MLE) is defined as:

$$\mathcal{L}_{\text{MLE}} = -\sum_{i=1}^{N}\sum_{t=1}^{T} p(y_t^i|y_1^i, ..., y_{t-1}^i, x^i) \quad (1)$$

3.2 Reinforcement Learning for NMT

Within the RL framework, the task of NMT can be formulated as a sequential decision making process, where the **state** is defined by the previously generated words ($\mathbf{y}_{<t}$) and the **action** is the next word to be generated. Given the state s_t, the agent picks an action a_t (for seq2seq it is the same as y_t), according to a (typically stochastic) **policy** π_θ and observes a reward r_t for that action. The reward can be calculated based on any evaluation metric, *e.g.* BLEU.

The objective of the RL training is to maximise the expected reward:

$$\mathcal{L}_{RL} = \mathbb{E}_{a_1,\cdots,a_T\sim\pi_\theta(a_1,\cdots,a_T)}[r(a_1, \cdots, a_T)] \quad (2)$$

Under the policy π, we can also define the values of the state-action pair $Q(s_t, y_t)$ and the state $V(s_t)$ as follows:

$$\begin{aligned} Q_\pi(s_t, a_t) &= \mathbb{E}[r_t|s = s_t, a = a_t] \\ V_\pi(s_t) &= \mathbb{E}_{a\sim\pi(s)}[Q_\pi(s_t, a = a_t)] \end{aligned} \quad (3)$$

Intuitively, the value function V measures how good the model could be when it is in a specific state s_t. The Q function measures the value of choosing a specific action when we are in such state.

Given the above definitions, we can define a function called *advantage* – denoted by A_π – relating the value function V and Q function as follows:

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t) \quad (4)$$

Therefore, the focus is on maximising one of the following objectives:

$$\max_a A_\pi(s_t, a_t) \rightarrow \max_a Q_\pi(s_t, a_t) \quad (5)$$

Different RL algorithms have different ways to search for the optimal policy. Algorithms such as REINFORCE, as well as its variant MIXER (Ranzato et al., 2015), popular in language tasks, search for the optimal policy via Eq. 2 using the Policy Gradient. Actor-Critic (AC) models typically improve the performance of Policy Gradient models by solving Eq. 5 (left part) (Bahdanau et al., 2016). Q-learning models that aim at maximising the Q function (Eq 5, right part) to improve over both the Policy Gradient and AC models (Dai et al., 2018).

3.3 Soft Actor-Critic (SAC)

The SAC algorithm (Haarnoja et al., 2018a) adds to the Eq. 2 an entropy term:

$$\mathcal{L}(\pi) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{a_t\sim\pi_{(\cdot|s_t)}} [r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))] \quad (6)$$

where α controls the stochasticity of the optimal policy, a trade-off between the relative importance of the entropy term $\mathcal{H}$ and the reward $r(s_t, a_t)$ that the agent receives by taking action a_t when the state of the environment is s_t. Its aim is to maximise the

entropy of actions at the same time as maximising the rewards.

As mentioned earlier, SAC is an off-policy Q-learning AC algorithm. As other AC algorithms it consists of two parts: the actor (the policy function) and the critic – action-value function (Q), parameterised by ϕ and θ, respectively.

During off-policy learning, the history of states, actions and respective rewards are stored in a memory (D), *a.k.a.* the replay buffer.

- **Critic Training**

The Q-function estimates the value of an action at a given state based on its future rewards. The soft-Q value is computed recursively by applying a modified Bellman backup operator:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \mathop{\mathbb{E}}_{s_{t+1} \sim D}[V(s_{t+1})] \quad (7)$$

where

$$V(s_t) = \mathop{\mathbb{E}}_{a_t \sim \pi}[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t)] \quad (8)$$

is the expected future reward of a state and $\log(\pi(a_t|s_t))$ is the entropy of the policy.

The parameters of the Q-function are updated towards minimising the mean squared error between the estimated Q-values and the assumed ground-truth Q-value. The assumed ground-truth Q-values are estimated based on the current reward ($r(s_t, a_t)$) and the discounted future reward of the next state ($\gamma V_{\bar{\theta}}(s_{t+1})$). This mean squared error objective function of the Q network is as follows:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{s_t, a_t, r_t, s_{t+1} \sim D, a_{t+1} \sim \pi_\phi} \Big[\big(Q_\theta(a_t, s_t) - [r(s_t, a_t) + \gamma \mathop{\mathbb{E}}_{s_{t+1} \sim D}[V_{\bar{\theta}}(s_{t+1})]] \big)^2 \Big] \quad (9)$$

Note that the parameters of the networks are denoted as θ and $\bar{\theta}$ respectively. This is the best practice where the critic is modeled with two neural networks with the exact same architecture but independent parameters (Mnih et al., 2015).

The parameters of the target critic network ($Q_{\bar{\theta}}$) are iteratively updated with the exponential moving average of the parameters of the main critic network (Q_θ). This constrains the parameters of the target network to update at a slower pace toward the parameters of the main critic, which has been shown to stabilise the training process (Lillicrap et al., 2016).

Another advantage of SAC is the double Q-learning (Hasselt, 2010). In this approach, two Q networks for both of the main and the target critic functions are maintained. When estimating the current Q values or the discounted future rewards, the minimum of the outputs of the two Q networks is used. Thus the estimated Q values do not grow too large, which improves the policy training (Haarnoja et al., 2018a).

- **Actor Training**

SAC updates the policy to minimise the KL-divergence to make the distribution of $\pi_\phi(s_t)$ policy function look more like the distribution of the Q function:

$$\mathcal{L}_\pi(\phi) = \mathop{\mathbb{E}}_{s_t \sim D}[\pi_t(s_t)^T [\alpha \log(\pi_\phi(s_t)) - Q_\theta(s_t)]] \quad (10)$$

where softmax is used in the final layer of the policy to output a probability distribution over the actions.

We note that some versions of the SAC algorithm allow to automatically tune the α parameter so that while maximising the expected return, the policy should satisfy the minimum entropy criteria. In our experiments we however used a fixed α. Updating α during training resulted in too short sentences in the output.

Finally, we note that Eq. 10 does *not* simply add an entropy term to the standard Policy Gradient. The critic Q_θ trained by Eq. 9 additionally captures the *entropy from future steps*.

For more details on SAC for the discrete setting (like MT) we refer to Christodoulou (2019). For more formal details on the architecture, see Haarnoja et al. (2018a,b).

3.4 Reward functions

Below we define the reward functions we use in our SAC architecture.

Supervised BLEU reward: - `SAC BLEU` In the supervised setup, we employ the sequence-level BLEU score (Papineni et al., 2002) with add-1 smoothing (Chen and Cherry, 2014). As an additional length constraint at each time step, we deduct from the respective score the length penalty: $lp = |l_{\mathbf{y}} - l_{\hat{\mathbf{y}}}|$, where $\mathbf{y}$ is the reference translation. This penalty prevents longer translations that are not penalised by the brevity penalty of BLEU.

BLEU has been chosen in our study to ensure better comparability with the related work in RL MT traditionally using the BLEU reward (Bahdanau et al., 2016; Dai et al., 2018).

Unsupervised reward - `SAC unsuper` As discussed above, using automatic metrics as reward function can lead to a number of issues, *e.g.* reward sparsity, overfitting towards single reference. Moreover, designing a good reward can be challenging.

Inspired by recent work on the SAC algorithm in unsupervised RL (Eysenbach et al., 2018), we have designed an unsupervised reward that *balances the quality and diversity in the model search space*.

The pseudo-reward function we use is as follows:

$$r_z(\mathbf{x}, a) = \log q_\delta(z|\mathbf{x}, a) - \log p(z) \qquad (11)$$

where $p(z)$ is a categorical uniform distribution for a latent variable z.

$q_\delta(z|\mathbf{x}, a)$ is provided by a discriminator parametrised by a neural network. z is randomly assigned to a word sampled at each step from the actor distribution. The discriminator is a Bag-of-Words model that takes as input the encoded source sequence and the word itself to predict its z.

More intuitively, every time a word appears in the translation hypothesis for a source sentence (within the Bag-of-Words formulation) it is randomly assigned a certain value of z. The more times this word appears in the sampled hypotheses (for a given source) the closer will be $\log q_\delta(z|\mathbf{x}, a)$ to the uniform prior $p(z)$, hence reward $r_z(\mathbf{x}, a)$ will be close to 0. Thus, frequent translations will be suppressed and search for less frequent translations will be encouraged in order to receive a reward larger than 0.

Such a reward is less sparse than the traditional ones and is also dynamic which prevents memorising and overfitting.

4 Experimental Setup

4.1 Data

We perform experiments on the **Multi30K** dataset (Elliott et al., 2016)[1] of image description translations and focus on the English-German (EN-DE) and English-French (EN-FR) (Elliott et al., 2017) language directions. Following best practises, we use sub-word segmentation (BPE (Sennrich et al., 2016)) only on the target side of the corpus. The dataset contains 29,000 instances for training, 1,014 for development, and 1,000 for testing. We use **flickr2016** (**2016**), **flickr2017** (**2017**) and **coco2017** (**COCO**) test sets for model evaluation.

2016 is the most **in-domain** test set since it was taken from the same superset of descriptions as the training set, whereas **2017** and **COCO** are from different image description corpora and are thus considered **out-of-domain**.

For more fine-grained assessment of our models with unsupervised reward, we use the **MLT** test set (Lala and Specia, 2018; Lala et al., 2019), an annotated subset of the **Multi30K** corpus where each instance is a 3-tuple consisting of an **ambiguous** source word, its textual context (a source sentence), and its correct translation. The test set contains 1,298 sentences for English-French and 1,708 for English-German. It was designed to benchmark models in their ability to select the right lexical choice for words with multiple translations, especially when some of these translations are rarer.

Additionally, to allow for comparison with previous work, we evaluate on the **IWSLT 2014** German-to-English dataset (Cettolo et al., 2012) from TED talks, which has been used as testbed in most work on RL for MT. The training set contains $153K$ sentence pairs. We followed the pre-processing procedure described in (Dai et al., 2018).

When compared to the **IWSLT 2014** dataset, all the three **Multi30K** test sets are more out-of-domain. This was found by the analysis of perplexities of language models trained with respective training data for each dataset (see Appendix A.4).

4.2 Training

We modify the original SAC architecture to adapt it to MT following best practices (Bahdanau et al., 2016) in the area. The functions π_ϕ and Q_θ are parameterised with neural networks: π_ϕ is an RNN seq2seq model with a 2-layer GRU (Cho et al., 2014) encoder and a 2-layer Conditional GRU decoder (Sennrich et al., 2017) with attention (Bahdanau et al., 2014). For `SAC BLEU`, Q_θ duplicates the structure of the former, but encodes the reference instead of the source sentence to mimic inputs to the actual BLEU function.

We first pretrain the actor and then pretrain the critic, before the actor-critic training. The pretraining of actors is done until convergence according

[1] https://github.com/multi30k/dataset

	model	2016			2017			COCO		
		BLEU	METEOR	TER	BLEU	METEOR	TER	BLEU	METEOR	TER
EN-FR	MLE	57.5	71.7	27.5	50.9	66.8	33.0	42.8	61.5	37.3
	ERAC (ours)	**59.4***	**73.3***	**26.7***	51.2	66.8	32.5	42.5*	60.6*	37.6*
	SAC BLEU	57.9	72.0	27.8	**51.7***	**67.5***	**32.1***	**44.4**	**62.9**	**36.4**
	SAC unsuper	56.9	71.4	28.2	51.1	67.1	32.5	43.6	62.6*	36.6
EN-DE	MLE	38.5	**57.2**	42.2	**31.9**	**51.3**	49.5	27.2	46.7	55.2
	ERAC (ours)	**38.9***	56.1*	**41.9***	31.4*	49.6*	49.7*	25.0	44.0	56.0*
	SAC BLEU	38.1	56.8*	42.5	**31.9**	51.2	**49.1**	**27.7**	**47.0**	**54.5**
	SAC unsuper	38.0*	56.9	43.0*	31.6	50.8	49.7*	26.6	46.5	55.1

Table 1: Performance of SAC BLEU on the **Multi30K** test sets (EN-FR, EN-DE) trained on the **Multi30K** train set. * marks statistically significant changes (p-value ≤ 0.05) as compared to MLE. Bold highlights best results. ERAC (ours) indicates results obtained by us using the code openly provided by Dai et al. (2018).

to the early stopping criteria of 10 epochs wrt. to the MLE loss. We have also found that our critics require much less pretraining (3-5 epochs as compared to 10-20 epochs in general for AC architectures with the MSE loss). Also, to prevent divergence during the actor-critic training, we continue performing MLE training using a smaller weight λ_{mle}. We set α to 0.01. Following Haarnoja et al. (2018a), we rescale the reward to the value inverse to α. Note that we did not find it useful to add to SAC the smoothing objective minimising variance of Q-values (Bahdanau et al., 2016; Dai et al., 2018). We presume that the double Q-learning significantly contributes to the stability of the network and additional smoothing is not required.

For SAC unsuper, we parameterise q_δ by a 2-layer feed-forward neural network, which takes the source as encoded by the actor and a_t and outputs $q_\delta(z|\mathbf{x}, a)$. We set z to take one of 4 values.[2] For this unsupervised setting, we do not train a Q-function. We instead operate in the oracle mode and following (Keneshloo et al., 2018) define true Q-value estimates and use it to update our actor. Details on training are given in Appendix A. We use pysimt (Caglayan et al., 2020) with PyTorch (Paszke et al., 2019) v1.4 for our experiments.[3]

4.3 Evaluation

We use the standard set of MT evaluation metrics: BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014) and TER (Snover et al., 2006). We perform significance testing via bootstrap resampling using the Multeval tool (Clark et al., 2011).

For the lexical translation task, we measure the **Lexical Translation Accuracy (LTA)** score (Lala et al., 2019). The score provides an average estimation of how accurately the words have been translated. For each ambiguous word, a score of +1 is awarded if the correct translation of the word is found in the output translation; a score of 0 is assigned if a known incorrect translation is found, or none of the candidate words are found in the translation. We also propose a metric that not only rewards correctly translated ambiguous words, but also penalises words translated with the wrong sense: the **Ambiguous Lexical Index (ALI)**. ALI assigns -1 for wrong translations in the given context, whereas LTA simply does not reward them.

5 Results

5.1 Comparison to state-of-the-art

We first compare our SAC models against the MLE model (baseline) and ERAC[4] (state-of-the-art – SOTA) both trained and tested on the **Multi30K** data (Table 1). Compared to SAC, ERAC differs in that it uses the on-policy setting (i.e., using samples collected in real time). Our SAC algorithm is an off-policy algorithm and uses samples from the memory to promote generalisation.

We clearly observe the tendency of ERAC models to perform better on the more in-domain **2016** data (+1.9 BLEU, +1.6 METEOR, -0.8 TER

[2]This hyperparameter is tuned on the validation set. It typically varies from 2 to several hundreds in the related work (Haarnoja et al., 2018b).

[3]https://github.com/ImperialNLP/pysimt

[4]For ERAC, we present results that we reproduced ourselves using the code publicly provided by the authors. We had to perform several modifications to this code to make it conform recent deep learning framework software updates. The performance of this model is on pair with this reported by the authors.

	Model	2016			2017			COCO		
		BLEU	METEOR	TER	BLEU	METEOR	TER	BLEU	METEOR	TER
UNK	MLE	25.1	**29.1**	**49.9**	23.1	**27.7**	54.5	18.9	**25.8**	59.2
	SAC BLEU	**25.2**	28.9	50.1	**23.2**	27.5	54.5	**19.4**	25.5*	**58.3***
noUNK	MLE	34.4	37.8	40.4	31.6	**37.7**	46.4	25.9	34.2	50.6
	SAC BLEU	**34.9**	**38.0**	**40.0**	**32.1**	**37.6**	**45.9**	**28.3***	**34.5**	**48.6**

Table 2: Performance of SAC BLEU on **Multi30K** (German-English) trained on the **IWSLT 2014** train set. UNK indicates standard output containing the UNK symbol; noUNK – outputs with sentences containing UNK not taken into account. * marks statistically significant changes (p-value ≤ 0.05) as compared to MLE. Bold highlights best results.

against MLE for EN-FR) and the tendency of SAC BLEU models to outperform other models on more out-of-domain **2017** and **COCO** sets (+2.7 BLEU and +3.0 METEOR, -1.5 TER against ERAC on **COCO** for EN-DE).

SAC unsuper results are however worse than the baseline and SOTA. We focus thus on the investigation of SAC BLEU and come back to SAC unsuper in Section 5.2.

To further confirm our hypothesis that SAC reduces overfitting and performs better on the out-of-domain data, we train our models on the **IWSLT 2014** train set and test on the out-of-domain **Multi30K** test sets (in the reverse direction, German into English, Table 2).

We observe similar performance for complete set of outputs (including sentences with UNK tokens) for MLE and SAC BLEU. If the lines with UNK words are not taken into account,[5] we observe an improvement for the **2016** and **2017** test sets (+0.5 BLEU, +0.1 METEOR, -0.5 TER on average), and a much bigger improvement for the more out-of-domain **COCO** set (+2.5 BLEU, +0.3 METEOR, -2 TER on average). This confirms our hypothesis that SAC helps to reduce overfitting.

Finally, we compare SAC to the SOTA AC-base RL architectures, namely ERAC and AC, on the **IWSLT 2014** set that is commonly used for this task. Compared to SAC, AC differs in that it does not use entropy regularisation. We also provide the performance for the popular MIXER algorithm. Results are shown in Table 3.

In terms of the general performance, our SAC performs on pair with the MLE model. SAC BLEU even slightly lowers this score (-0.2 BLEU, -0.2 METEOR). We note that SAC BLEU results contain an increased count of UNK words as compared to MLE (+2.8%) This increased generation of UNK words due to the entropy regularisation is partially responsible for this similar performance. Another cause is that SAC does not overfit to the BLEU distribution of the target data.[6]

Model	BLEU	METEOR	TER
MLE (ours)	29.8	31.2	48.9
MIXER (Ranzato et al., 2015)	20.73	-	-
AC (Bahdanau et al., 2016)	28.53	-	-
ERAC (w/feed) (Dai et al., 2018)	29.36	-	-
ERAC (w/o feed) (Dai et al., 2018)	28.42	-	-
ERAC (w/o feed, ours)	29.0*	30.6*	51.5*
SAC BLEU	**29.6***	**31.0***	**48.8***

Table 3: Performance of MLE and different RL algorithms on the **IWSLT 2014** test set trained on the **IWSLT 2014** train set. * marks statistically significant changes (p-value ≤ 0.05) as compared to MLE. Bold highlights best RL results. MIXER, AC and ERAC scores were taken from original papers. ERAC (ours) indicates our results using the code provided in (Dai et al., 2018).

5.2 Translation of ambiguous words

To further investigate the effect of the unsupervised reward, we have evaluated SAC unsuper on the **MLT** dataset. Results are shown in Table 4. We calculate the scores on two conditions: **All Cases** takes into account all possible lexical translations; while for **Rare Cases**, only the instances where the gold-standard translation is not the most frequent translation for that particular ambiguous word. We observe that both SAC BLEU and SAC unsuper

[5]The original corpus pre-processing pipeline that we followed to increase comparability does not include subword segmentation. We take the intersection of hypotheses sentences across **Multi30K** test setups that contain no generated UNK token wrt. the **IWSLT 2014** vocabulary. Reference files may still contain the UNK token, we focus on the generated text here.

[6]We mean that the model would have a tendency to select certain words to simply boost BLEU rather than picking words to reflect the correct meaning.

	Model	All Cases						Rare Cases					
		2016		2017		COCO		2016		2017		COCO	
		LTA	ALI	LTA	ALI	LTA	ALI	LTA	ALI	LTA	ALI	LTA	ALI
EN-FR	MLE	81.60	63.19	79.65	59.31	74.60	49.21	52.81	24.49	**47.80**	**16.48**	47.16	**18.49**
	SAC BLEU	81.94	63.89	79.76	59.53	**77.32**	**54.65**	53.37	25.39	45.91	13.46	**49.05**	15.47
	SAC unsuper	**82.75**	**65.51**	**80.62**	**61.25**	75.28	50.57	**54.49**	**27.19**	**47.80**	**16.48**	47.16	15.47
EN-DE	MLE	65.34	30.68	70.91	41.82	67.45	34.91	50.95	11.72	60.00	28.00	56.56	21.82
	SAC BLEU	64.74	29.48	71.93	43.86	**67.72**	**35.43**	50.14	10.24	60.58	29.04	**58.58**	**25.45**
	SAC unsuper	**65.54**	**31.08**	**73.41**	**46.82**	66.40	32.81	**51.50**	**12.70**	**63.77**	**34.78**	52.52	14.55

Table 4: Performance of SAC BLEU on the **MLT** test sets (EN-FR, EN-DE). We report Ambiguous Words Accuracy: LTA and ALI. **Rare Cases** indicates the cases where the correct translation is *not* the most frequent translation in the training set.

outperform the MLE baseline across metrics in all setups except for the **COCO** EN-FR translation in Rare Cases, where MLE performs better. For SAC BLEU, this observation is also shown by general evaluation metrics BLEU, METEOR and TER on all **MLT** test sets (see Table 9 in Appendix).

Moreover, SAC unsuper is particularly successful when evaluated on **2016** and **2017** and outperforms both MLE and SAC BLEU across setups. This demonstrates the potential of the unsupervised reward function for the cases when we have to choose between possible translations for an ambiguous word (i.e., better exploration of the search space). BLEU reward, on the other hand, is more reliable when we have to adjust distributions to produce one single possible translation. Manual inspection of these SAC unsuper improvements confirmed their increased accuracy (see Table 5). For example, the ambiguous French source word 'hill' ('colline') is translated as 'pente'('slope') by both MLE and SAC BLEU, while only SAC unsuper produces the correct sentence: 'adolescent saute la *colline* 'hill' avec son vélo'.

5.3 Qualitative analysis

To get further insights into the general results, we also performed human evaluation of the outputs for MLE, SAC BLEU, and SAC unsuper using professional in-house expertise. This was done for **COCO** EN-FR and **2016** EN-DE as two sets with contrastive results in the lexical translation experiment.

For this human analysis, we randomly selected test samples (50 samples per language pair per group) with source words of different frequency in the training data: rare words (frequency 1) and other words (frequency ≥ 10). These other words are randomly chosen from the sentences that differ in their translation across setups. The resulting average frequency of those words is around 40 for both language pairs. A rank of quality (both fluency and adequacy together) is assigned by the human evaluator from 1 to 3, allowing ties. Following the common practice in MT, each system was then assigned a score which reflects how often it was judged to be better or equal to other systems (Bojar et al., 2017).

Results are in Table 6. We observe a tendency of SAC BLEU to do well on the translation of rare source words, but not so well on the translation of words in the middle frequency range (this observation is confirmed by the analysis of the frequency of output words, see Appendix A.5, see Table 10). Our unsupervised reward tends to increase the performance on more frequent words ('Other' in Table 6) by promoting their less common translations in the distribution, hence better translations for ambiguous words from our previous experiment. These ambiguous words are quite frequent, they potentially have multiple possible translations but only one correct translation in a given context.

6 Conclusions

We propose and reformulate SAC reinforcement learning approaches to help machine translation through better exploration and less reliance on the reward function. To provide a good trade-off between exploration and quality, we devise two reward methods in the supervised and dynamic unsupervised manner. The maximum entropy off-policy SAC algorithm mitigates the overfitting problem when evaluated in the out-of-domain space; both rewards introduced in our SAC architecture can achieve better quality for lexical translation of ambiguous words, particularly the rare senses of

EN-FR	source word	hill
	gold target word	colline
	source sentence	the teen jumps the **hill** with his bicycle .
	reference sentence	ado saute sur la **colline** 'hill' avec son vélo .
	MLE	adolescent saute sur la **pente** 'slope' avec son vélo .
	SAC BLEU	adolescent saute la **pente** 'slope' avec son vélo .
	SAC unsuper	adolescent saute la **colline** 'hill' avec son vélo .
EN-DE	source word	outfit
	gold target word	outfit
	source sentence	a rhythmic gymnast in a blue and pink **outfit** performs a ribbon routine .
	reference sentence	eine rhythmische sportgymnastin in einem blauen und pinken **outfit** vollführt eine bewegung mit dem band .
	MLE	ein begeisterter turner in blau-rosa **kleidung** 'dress' führt eine band auf .
	SAC BLEU	ein begeisterter turner in blau-rosa **kleidung** 'dress' führt eine band auf .
	SAC unsuper	ein aufgeregter turner in einem blau-rosa **outfit** führt eine band aus .

Table 5: Samples of ambiguous words translation on **2016** for both EN-FR and EN-DE. In both cases more correct translations are provided by SAC unsuper. Bold highlights target words and their translations.

Lang	Words	MLE	SAC BLEU	SAC unsuper
EN-FR	Rare (Freq. 1)	1.76	**1.88**	1.68
	Other	**1.88**	1.82	1.86
EN-DE	Rare (Freq. 1)	1.72	**1.74**	1.70
	Other	1.93	1.83	**1.94**

Table 6: Human ranking results for **2016** EN-DE and **COCO** EN-FR test set. Bold highlights best results per group of word types. The first column indicates the groups of word types. Results are averaged for all words per word type group.

words. The formulation of the unsupervised reward and its potential to influence translation quality open perspectives for future studies on the subject. We leave the exploration of how those supervised and unsupervised rewards could be combined to improve MT for future work.

Acknowledgments

The authors thank the anonymous reviewers for their useful feedback. This work was supported by the MultiMT (H2020 ERC Starting Grant No. 678017) and the Air Force Office of Scientific Research (under award number FA8655-20-1-7006) projects. Marina Fomicheva and Lucia Specia were supported by funding from the Bergamot project (EU H2020 grant no. 825303). We also thank the annotators for their valuable help.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ozan Caglayan, Julia Ive, Veneta Haralampieva, Pranava Madhyastha, Loïc Barrault, and Lucia Specia. 2020. Simultaneous machine translation with visual context. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2350–2361, Online. Association for Computational Linguistics.

Mauro Cettolo, C. Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. *Proceedings of EAMT*, pages 261–268.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. On the weaknesses of reinforcement learning for neural machine translation. In *International Conference on Learning Representations*.

Petros Christodoulou. 2019. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

Zihang Dai, Qizhe Xie, and Eduard Hovy. 2018. From credit assignment to entropy regularization: Two new algorithms for neural sequence prediction. *arXiv preprint arXiv:1804.10974*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. 2018. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online. Association for Computational Linguistics.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. *CoRR*, abs/1702.08165.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018a. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. 2018b. Soft actor-critic algorithms and applications. *CoRR*.

Hado V. Hasselt. 2010. Double q-learning. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2613–2621. Curran Associates, Inc.

Di He, Hanqing Lu, Yingce Xia, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2017. Decoding with value networks for neural machine translation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 178–187. Curran Associates, Inc.

Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K. Reddy. 2018. Deep reinforcement learning for sequence to sequence models. *arXiv preprint arXiv:1805.09461*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Vijay R. Konda and John N. Tsitsiklis. 2000. Actor-critic algorithms. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 1008–1014. MIT Press.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. 2018. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Melbourne, Australia. Association for Computational Linguistics.

Chiraag Lala, Pranava Madhyastha, and Lucia Specia. 2019. Grounded word sense translation. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language*, pages 78–85, Minneapolis, Minnesota. Association for Computational Linguistics.

Chiraag Lala and Lucia Specia. 2018. Multimodal Lexical Translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *ICLR*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. pages 1433–1438. AAAI Press.

A Training Details

A.1 Hyperparameters

For the NMT RNN agent, the dimensions of embeddings and GRU hidden states are set to 200 and 320, respectively. The decoder's input and output embeddings are shared (Press and Wolf, 2017). We use Adam (Kingma and Ba, 2014) as the optimiser and set the learning rate and mini-batch size to 0.0004 and 64, respectively. A weight decay of $1e{-}5$ is applied for regularisation. We clip the gradients if the norm of the full parameter vector exceeds 1 (Pascanu et al., 2013). The four Q-networks are identical to the agent (see Table 7).

For the unsupervised reward setting, we use 2 two-layer feed-forward neural network (both dimensionalities are equal to 100). We use again Adam as the optimiser and set the learning rate and mini-batch size to 0.0001 and 64, respectively.

Hyper-parameters	
Pre-train Critic	
optimiser	Adam
learning rate	0.0003
batch size	64
τ (target net speed)	0.005
α (entropy regularization)	0.001
buffer size	1000
length penalty	0.0001
Joint Training	
optimiser	Adam
learning rate	0.0004
batch size	64
τ (target net speed)	0.005
α (entropy regularization)	0.001
buffer size	1000
length penalty	0.0001
λ_{MLE}	0.1

Table 7: Hyper-parameters for SAC training.

A.2 Training

We use PyTorch (Paszke et al., 2019) (v1.4, CUDA 10.1) for our experiments. We early stop the actor training if validation loss does not improve for 10 epochs, we pretrain critics for 5 epochs for the **Multi30K** datasets and for 3 epochs for the larger **IWSLT 2014**. We early stop the SAC training if validation BLEU does not improve for 10 epochs. For all the setups, we also halve the learning rate if no improvement is obtained for two epochs. On a single NVIDIA RTX2080-Ti GPU, it takes around 5-6 hours up to 36 hours to train a model depending on the data size and the language pair. The number of learnable parameters is about 7.89M for smaller Multi30K models and about for 15.64M for the bigger IWSLT model. All models were re-trained 3 times to ensure reproducibility.

A.3 Soft Actor-Critic Training Algorithm

We describe the main steps of SAC training in Algorithm 1.

Algorithm 1: Soft Actor-Critic.

Initialise parameters:
Q function: θ;
Policy: ϕ;
Unsupervised Reward: δ;
Replay Buffer: $\mathcal{D} \leftarrow \emptyset$;
for *each iteration* **do**
  for *each translation step* **do**
    $a_t \sim \pi_\phi(a_t, s_t)$;
    $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$;
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_t, a_t, r(s_t, a_t), s_{t+1}\}$;
  end
  for *each gradient step* **do**
    $\theta_i \leftarrow \theta_i - \lambda_Q$
     $\nabla_{\theta_i} L(\theta_i)$ for $i \in \{1, 2\}$;
    $\phi \leftarrow \phi - \lambda_\pi \nabla_\phi J(\phi)$;
    $\alpha \leftarrow \alpha - \lambda_\pi \nabla_\alpha J(\alpha)$;
    $\theta_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i$
    for $i \in \{1, 2\}$;
    if *unsupervised reward* **then**
      $\delta \leftarrow \delta - \lambda_z \nabla_\delta r(\delta)$;
    end
  end
end

LM	**2016**	**2017**	**COCO**
Multi30K	44.07	79.95	77.7
IWSLT 2014	579.47	403.54	381.56

Table 8: Perplexity on **Multi30K** testsets for **Multi30K** and **IWSLT 2014** language models.

A.4 Domain Distance

To assess to what extent the test sets used in our experiments can be considered out-of-domain, we train (i) an English language model on **Multi30K** training set; and (ii) a German language model on

	model	2016			2017			COCO		
		BLEU	METEOR	TER	BLEU	METEOR	TER	BLEU	METEOR	TER
EN-FR	MLE	58.8	73.8	**26.7**	54.2	70.2	30.1	42.6	62.1	36.0
	SAC BLEU	**59.4**	**74.0**	**26.7***	**55.2**	**70.8**	**29.2**	**44.1**	**63.4**	**35.5**
	SAC unsuper	58.2	73.6	27.3	54.4*	70.6	29.8	43.5	63.2	35.7*
EN-DE	MLE	**37.5**	56.3	**42.1**	**33.8**	**53.1**	**47.6**	29.3	**49.3**	**50.9**
	SAC BLEU	36.6	56.2*	43.2	33.5	**53.1***	**47.6***	**29.6***	**49.3***	51.0*
	SAC unsuper	36.3	**56.5***	44.1	33.1	52.9*	48.7	28.3	48.6	51.5

Table 9: Results on the test sets for ambiguous words.

Freq. 1	source word	traveler
	gold target word	reisender
	source sentence	an oriental **traveler** awaits his turn at the currency exchange .
	reference sentence	ein orientalischer **reisender** ‘traveler’ wartet am wechselschalter bis er dran ist .
	MLE	ein orientalisch aussehender **behinderter** ‘disabled’ wartet darauf , dass die kurve sich die glastür aufhebt .
	SAC BLEU	ein orientalisch aussehender **techniker** ‘technician’ wartet auf die hecke seiner kurve .
	SAC unsuper	ein orientalisch aussehender **mann** ‘man’ wartet darauf , dass seine kurve auf den fehenk die kurve ist .
Freq. 28	source word	check
	gold target word	scheck
	source sentence	a woman is holding a large **check** for kids food basket .
	reference sentence	eine frau hält einen großen **scheck** ‘check’ für ” kids’ food basket ” .
	MLE	eine frau hält ein großes **überprüfen** ‘proof’ für kinder .
	SAC BLEU	eine frau hält einen großen **informationen** ‘information’ für kinder in den korb .
	SAC unsuper	eine frau hält ein großes **überprüfen** ‘proof’ für kinder , die einen korb zu verkaufen ist .

Table 10: Samples of translations for words of different frequency on **2016** EN-DE. In both cases more correct translations are provided by SAC unsuper. Bold highlights target words and their translations.

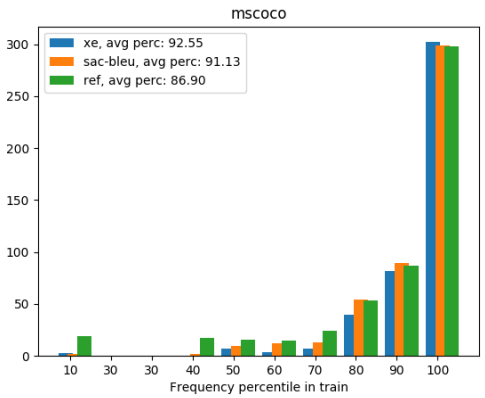

Figure 1: Training frequency for **COCO** words as translated by MLE and SAC BLEU. We also report reference frequencies.

the **IWSLT 2014** training set.[7] Table 8 shows language model perplexities on the Mutli30k test data. With respect to the **IWSLT 2014** model, **Multi30K** test sets are clearly very different from the training data. With respect to the Multi30K model, **2017** and **COCO** are more distant from the train partition than 2016 testset.

[7]We train Transformer language models using the fairseq toolkit (Ott et al., 2019).

A.5 Analysis of distributions

We argue that the improvement over MLE can be partially attributed to a better handling of less frequent words. It has been shown that rare words tend to be under-represented in NMT (Koehn and Knowles, 2017; Shen et al., 2016). RL training with regularized entropy might mitigate this issue due to a better exploration of the action space. To illustrate this point, we compute the training frequency of the words generated by the NMT systems for the sentences where an improvement over MLE is observed. Figure 1 shows the training frequency percentiles for MLE and SAC BLEU English-French translations of the **COCO** testset. Reference frequencies are also provided for comparison. We observe that although both MLE and SAC contain more frequent words than the reference, this tendency is less pronounced for SAC. We relate this observation to the fact that our SAC outperforms MLE for the ambiguous word translation (Table 4) where the most frequent translation is not always the correct one.